

一、transition 过渡

1 过渡的属性 transition-property

- 如果想要写多个属性，利用逗号进行分割 transition-property: width, height;
- all可以作用全部需要过渡的属性 transition-property: all;
- 当前元素一定要存在该过渡属性

2 过渡时间 transition-duration (在多长的时间内完成过渡)

- transition-duration: 2s;

3 过渡曲线 transition-timing-function

- linear 匀速
- ease-in 先慢后快
- ease-out 先快后慢
- ease-in-out 前后慢中间快
- cubic-bezier(.83,.29,.01,.79) 贝塞尔曲线 cubic-bezier.com
- transition-timing-function: cubic-bezier(.83,.29,.01,.79);

4 过渡延时 transition-delay

- transition-delay: 3s;

5 简写

- transition: 过渡属性 过渡时间 过渡曲线 过渡延时;

```
1 transition: all 10s ease-in 3s;
```

6 原则：谁要过渡给谁加

```
1 <style>
2   div {
3     width: 200px;
4     height: 100px;
5     background-color: pink;
6     /* transition: 变化的属性 花费时间 运动曲线 何时开始; */
7     /* 如果想要写多个属性，利用逗号进行分割 */
8     transition: width 0.5s, height 0.5s;
```

```

9      /* 如果想要多个属性都变化，属性写all就可以了 */
10     transition: all 0.5s;
11 }
12
13 div:hover {
14     width: 400px;
15     height: 200px;
16     background-color: red;
17 }
18 </style>

```

二、CSS3动画

1. 动画 **animation** 是CSS3中具有颠覆性的特征之一，可通过设置多个节点来精确控制一个或一组动画，常用来实现复杂的动画效果。
2. 相比较过渡，动画可以实现更多变化，更多控制，连续自动播放等效果。
3. 制作动画分为两步：
 - 先定义动画
 - 再使用（调用）动画

1 用keyframes定义动画

用 **keyframes** 定义动画

```

1  @keyframes 动画名称 {
2      0%{
3          width:100px;
4      }
5      100%{
6          width:200px;
7      }
8  }

```

- 0% 是动画的开始，100% 是动画的完成。这样的规则就是动画序列。
- 在 @keyframes 中规定某项 CSS 样式，就能创建由当前样式逐渐改为新样式的动画效果
- 动画是使元素从一种样式逐渐变化为另一种样式的效果。您可以改变任意多的样式任意多的次数。
- 请用百分比来规定变化发生的时间，或用关键词 “from” 和 “to” ，等同于 0% 和 100%。

2 使用动画

```

1  div {
2      width: 200px;
3      height: 200px;
4      background-color: aqua;
5      margin: 100px auto;
6      /* 调用动画 */
7      animation-name: 动画名称;
8      /* 持续时间 */
9      animation-duration: 持续时间;
10 }
11 @keyframes 动画名称 {
12     0%{
13         width:100px;
14     }
15     100%{
16         width:200px;
17     }
18 }

```

3 eg:

```

1  <style>
2      * {
3          margin: 0;
4          padding: 0;
5      }
6
7      .demo {
8          width: 632px;
9          height: 340px;
10         margin: 200px auto;
11         background: url(./img/g1.jpg);
12         animation: move 20s infinite normal forwards;
13
14     }
15
16     @keyframes move {
17         from {

```

```
18         background: url(/img/g1.jpg);
19     }
20
21     45% {
22         background: url(/img/g2.jpg);
23     }
24
25     50% {
26         background: url(/img/g2.jpg);
27     }
28
29     95% {
30         background: url(/img/g3.jpg);
31     }
32
33     to {
34         background: url(/img/g3.jpg);
35     }
36 }
37 </style>
38 </head>
39
40 <body>
41     <div class="demo">
42     </div>
43 </body>
```



4 常用属性

	描述
@keyframes	规定动画。
animation	所有动画属性的简写属性，除了animation-play-state属性。
animation-name	规定@keyframes动画的名称。（必须的）
animation-duration	规定动画完成一个周期所用时间（s,ms），默认是0。（必须的）
animation-timing-function	规定动画的速度曲线，默认是“ease”。 linear 匀速 ease-in 先慢后快 ease-out 先快后慢 ease-in-out 前后慢中间快 cubic-bezier 贝塞尔曲线 cubic-bezier.com
animation-delay	规定动画延时开始，默认是0。
animation-iteration-count	规定动画被播放的次数，默认是1；infinite（无限循环）
animation-direction	规定动画是否交替播放，默认是“normal”， normal from到to reverse 到到from alternate 逆交替播放 从from到to再从to到from，次数必须2次以上 alternate-reverse 交替反转 从to到from再从from到to

animation-play-state	规定动画是否正在运行或暂停。默认是"running",还有"paused"。
animation-fill-mode	规定动画结束后状态，保持forwards回到起始 backwards forwards 以默认状态开始 to里面的状态结束 backwards 以from里面的状态开始 默认状态结束 both 以form里面的状态开始 to里面的状态结束

5 动画简写属性

animation: 动画名称 持续时间 运动曲线 何时(延时)开始 播放次数 是否交替 动画起始或者结束的状态

```
1 animation: move 5s linear 2s infinite alternate;
```

- 简写属性里面不包含 animation-play-state
- 暂停动画: animation-play-state: paused; 经常和鼠标经过等其他配合使用
- 想要动画走回来，而不是直接跳回来: animation-direction: alternate
- 盒子动画结束后，停在结束位置: animation-fill-mode : forward;

三、layui

- 第一步引入 layui 文件夹下面的css文件
- 第二步 在body里面写script标签，script标签里面的src属性引入layui文件下的js文件

```
1 <head>
2   <meta charset="UTF-8">
3   <meta http-equiv="X-UA-Compatible" content="IE=edge">
4   <meta name="viewport" content="width=device-width, initial-scale=1.0">
5   <title>Document</title>
6
7   <!-- 第一步 引入 layui 文件夹下面的css文件 -->
8   <link rel="stylesheet" href="./layui/css/layui.css">
9 </head>
10 <body>
11
12   <i class="layui-icon">&#xe60c;</i>
13   <i class="layui-icon" style="color: red;">&#xe680;</i>
14
```

```
15 <!-- 第二步 在body里面写script标签，script标签里面的src属性引入layui文件下的js文件 -->
16 <script src="../layui/layui.js"></script>
17 </body>
```

7 媒体查询

```
1 <style>
2     * {
3         margin: 0;
4         padding: 0;
5     }
6     /*
7     媒体查询
8     @media 声明媒体查询
9     媒体类型 screen
10    关键词 and
11    限制条件 max-width min-width
12
13    */
14
15    @media screen and (max-width:900px) and (min-width:600px) {
16        body {
17            background-color: pink;
18        }
19    }
20 </style>
```